



RAUL BAREÑO GUTIERREZ

# Introducción

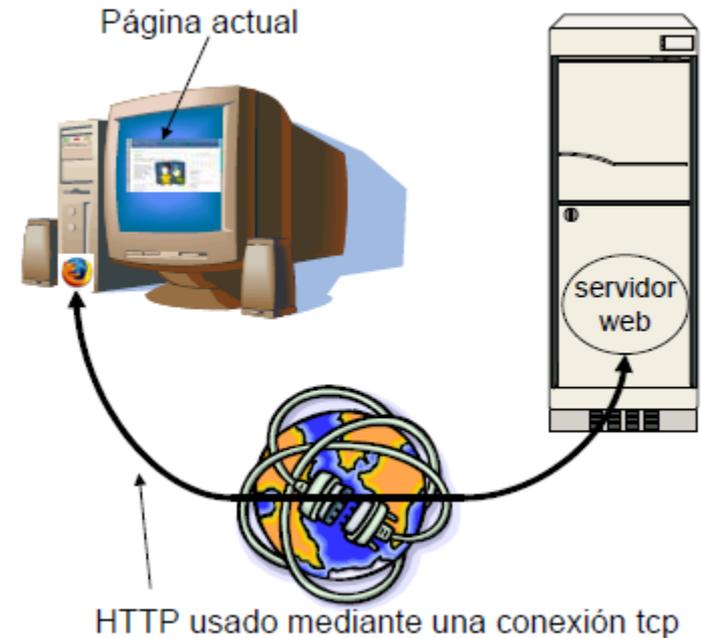
- Propuesto por Tim Berners-Lee en 1989
- HTTP es *utilizado para intercambio de Información* entre clientes Web (ej. Mozilla Firefox) y Servidores HTTP (ej. Apache)
- **Conexión TCP puerto 80** que *escucha pasivamente*.
- Actualmente versión HTTP/1.1
- HTTP/1.0 ver *RFC 1945* y HTTP/1.1 ver *RFC 2616*

# Funcionamiento

- 1. El cliente realiza una petición o apertura activa (**request**) al servidor (**puerto 80, por defecto**)
- 2. Solicita la transacción con HTTP: GET, POST, HEAD, PUT, ...
- 3. El servidor envía la respuesta (**response**) en HTML
- 4. Se cierra la conexión (en HTTP/1.0)
  - (Protocolo para la Transferencia de HiperTexto)
  - HTTP es el conjunto de reglas para la transmisión y recepción de documentos hipertexto. es el protocolo responsable del entendimiento universal de las páginas WWW escritas en HTML.
  - Una debilidad del HTTP es que carece de seguridad para la información transmitida.
  - **HTTP Seguro (*Secure HTTP*)**, goza de bastante popularidad.

## Ejemplo de transacción

- El navegador solicita recurso
- Se determina la URL
- Se resuelve la IP (DNS)
- Se establece conexión TCP con puerto 80 de la IP destino
- Se transmite el método GET <URI> <protocolo> (/archivo.html HTTP/1.1)
- El servidor responde (según extensiones MIME y RFC822)
- Se cierra la conexión (HTTP 1.0)
- Se presenta el recurso en el “navegador”



# Mensajes HTTP

- Dos tipos de mensajes usando ASCII (texto plano): **Request (Solicitud)**
- **Response (Respuesta)**
- **Solicitudes HTTP/1.0**

## ■ Solicitudes HTTP/1.0

Línea de solicitud → `GET /path/pagina.html HTTP/1.0`  
 (método, URI, ver.)  
 Líneas de cabecera → `User-agent: Mozilla/6.0`  
                           `Accept: text/html, image/gif, image/jpg`  
                           `Accept-language: es`  
 Fin de la cabecera → `<CR><LF>`  
  
 Datos (opcional) →

## ■ Respuestas HTTP/1.0

Línea de estado → `HTTP/1.0 200 OK`  
 (ver., código, frase)  
 Líneas de cabecera → `Date: Thu, 23 Nov 2004 12:00:15GMT`  
                           `Server: Apache/2.0 (Unix)`  
                           `Last-Modified: Mon, 10 Sept 2004`  
                           `...`  
                           `Content-Length: 6821`  
                           `Content-Type: text/html`  
                           `...`  
 Datos → `<html> ... .. </html>`

# Mensajes HTTP

- Códigos de estado:
  - Identificador del estado de la petición.
  - Los envía el servidor web como respuesta.
  - Entero de 3 dígitos:
    - 1xx: Informativos.
    - 2xx: Operación realizada con éxito.
    - 3xx: Redireccionan al cliente a otra URL.
    - 4xx: Error del cliente.
    - 5xx: Error del servidor.
  - Los más usuales:
    - 200 OK: solicitud exitosa, la respuesta va en el cuerpo.
    - 404 Not Found: el recursos no existe.
    - 303 See Other: el recurso se ha movido a otra URL (ver Header Location).
    - 500 Server Error: error interno del servidor.

# métodos

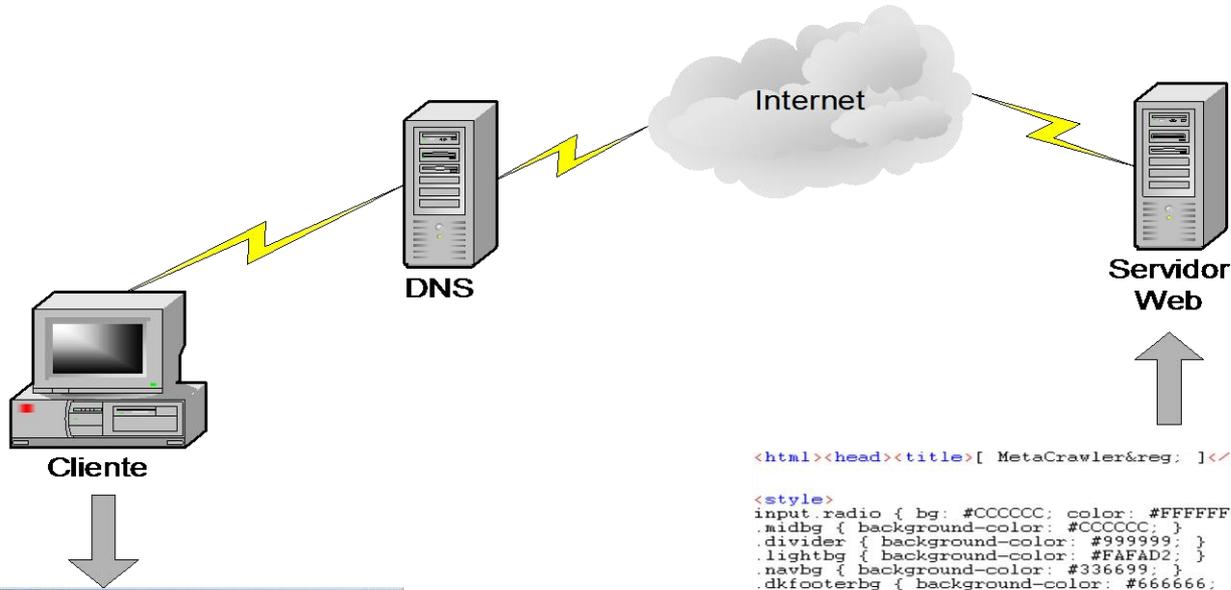
- Métodos:
  - Operaciones que se pueden solicitar a un servidor.
  - Existen 3 métodos distintos:
    - GET: Se utiliza para recoger cualquier tipo de información del servidor. Esta información va en el cuerpo de la respuesta.
    - HEAD: Solicita información sobre un objeto (fichero). Es igual que GET, pero la información va contenida en la cabecera de la respuesta. NUNCA TIENEN CUERPO.
    - POST: Se utiliza para enviar información al servidor, por ejemplo los datos contenidos en un formulario.

# APLICACIONES WEB

- Es un Conjunto de Páginas WEB que Interactúan entre sí.
- Una Aplicación WEB es un Sitio WEB que Contiene Páginas Almacenadas en un Servidor WEB.
- Se Crean Como Respuesta a Diversas Necesidades.
- **NO** Hay Que Confundir una **Aplicación WEB** con un **Portal o Página WEB** (Vista del Aspecto Solamente de que Muestra Información).



# WORLD WIDE WEB



```
<html><head><title>[ MetaCrawler&reg; ]</title>

<style>
input.radio { bg: #CCCCCC; color: #FFFFFF; }
.midbg { background-color: #CCCCCC; }
.divider { background-color: #999999; }
.lightbg { background-color: #FAFAD2; }
.navbg { background-color: #336699; }
.dkfooterbg { background-color: #666666; }</style>

<meta name="Description" content="Searching for something? MetaCrawler -
<meta name="Keywords" content="search, engine, MetaCrawler, Go2Net, mult
<body link="#0033cc" alink="#cccc99" vlink="#666666" text="#000000" bgcc

<div align="center">
<form name="querybox" action="http://search.metacrawler.com/crawler" met
<table cellpadding=0 cellspacing=0 width=594 border=0><tr><td align="rig
<a href="http://www.go2net.com" target="_top" onMouseOver="window.status
</td></tr>
</table>
<table cellpadding=0 cellspacing=0 border=0 width=594>
<!-- START tabs -->
<tr><td height="42" colspan="3">

<head>
<title>Instituto de Empresa</title>
<meta name="description" content="Escuela de Negocios Privada e Independiente Fundada en
1973">
<meta name="keywords"
content="MBA,Master, Seminarios,Cursos, Conferencias, Negocios, Escuelas, Programas, España, Direc-
ción, Educación">
<script language="JavaScript">

    var activado=new Image();
    activado.src="22A.JPG";
    var desactivado= new Image();
    desactivado.src="22.jpg" ;
    var activado2=new Image();
    activado2.src="22D.JPG";

    function activar(prueba) {
        document[prueba].src=activado.src;
    }
    function activar2(prueba) {
        document[prueba].src=activado2.src;
    }
    function desactivar(prueba) {
        document[prueba].src=desactivado.src;
    }
</script>
</head>
<script language="JavaScript">

<!--

```

## En el fondo, Cliente-Servidor ...

- Back End:
  - Servidores web:
    - Escuchan, esperando conexiones de clientes.
    - Envían vía HTTP la información que compone las páginas.
    - Dan a los navegadores las instrucciones precisas para presentar la información vía HTML
- Front End:
  - Navegadores o Browsers:
    - Usan HTTP para conectar con los servidores.
    - Piden y presentan la información recibida
- **Cualquier máquina conectada a Internet puede ser cliente o servidor web. Sólo necesita el software.**

## Las URL's

- Por Universal Resource Locators.
- Definen:
  - Nombre del recurso (nombre de archivo).
  - Nodo de Internet en que se encuentra.
  - Protocolo de acceso al recurso.
- Ejemplos: <http://www.uis.edu.co/index.htm>
  - Protocolo HTTP
  - Dirección: www.uis.edu Nodo con IP 195.53.226.130
  - Archivo index.html
- <ftp://ftp.rediris.es/LEEME>
  - Protocolo FTP
  - Dirección pub.rediris.es
  - Archivo LEEME

## Examinando una dirección web

**http:// www. ie . edu**

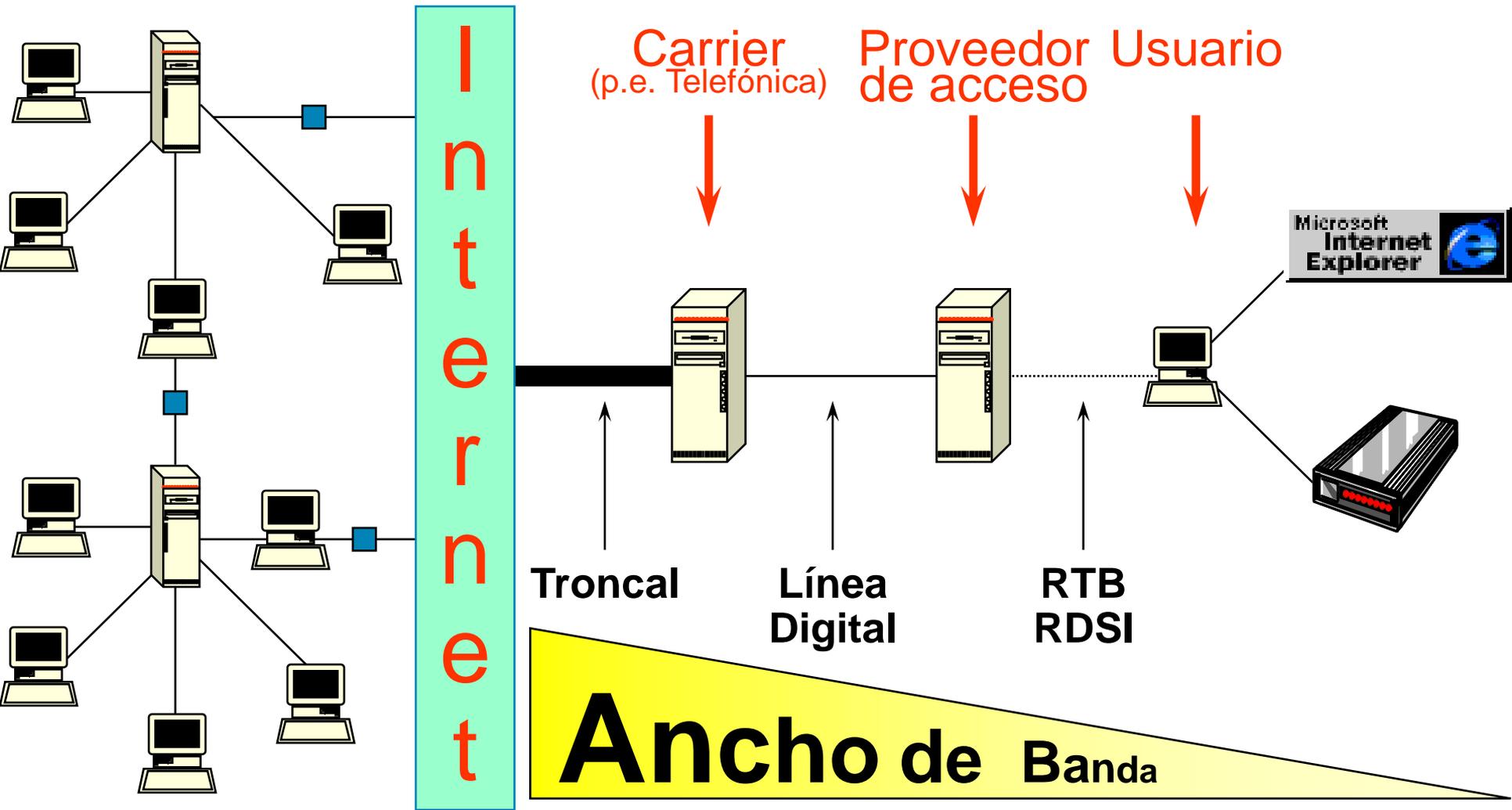
**Protocolo de Transferencia de Hipertexto**

**World Wide Web**

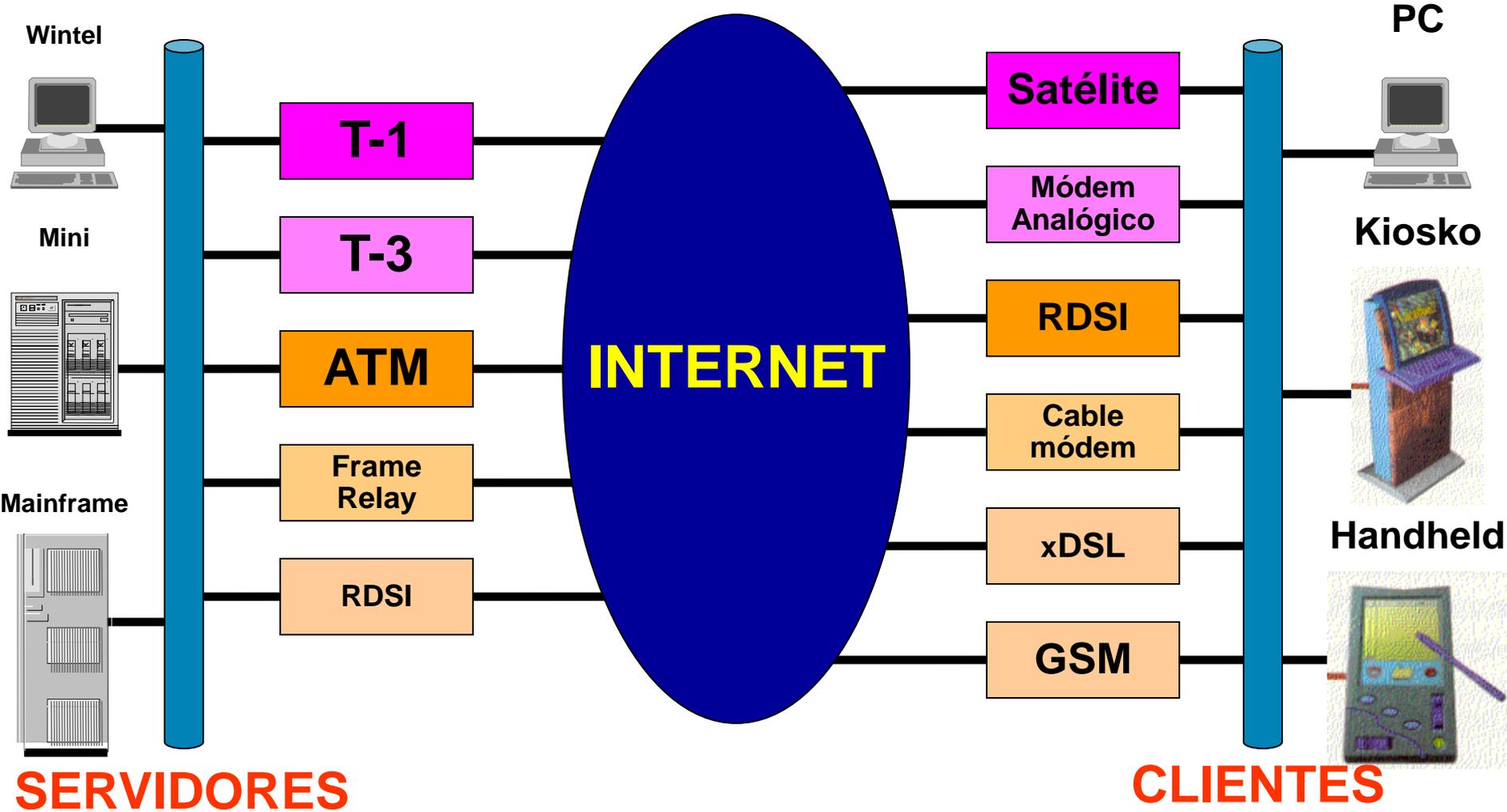
**Dominio**

**TLD**

# ¿Cómo accede un particular?



## Modos de conexión más habituales



**SERVIDORES**

**CLIENTES**

## Sesión típica de uso de WWW

- El usuario requiere la página [www.uis.edu/index.html](http://www.uis.edu/index.html)
  - El navegador pide al DNS la IP de [www.uis.edu](http://www.uis.edu)
    - DNS contesta 195.53.226.130
- El navegador hace una conexión TCP con 195.53.226.130
  - El navegador envía un comando GET /index.html
    - El servidor envía el fichero Index.html.
      - Se libera la conexión TCP.
    - El navegador muestra el texto de Index.html
- El navegador busca y muestra el resto de componentes de la página.



MUCHAS GRACIAS  
CONSTRUIMOS FUTURO

Cisco | Networking Academy<sup>®</sup>  
Mind Wide Open<sup>™</sup>